

REMARKS

Withdrawal of Appeal: In view of the filing of the Request For Continued Examination, the Appeal is hereby withdrawn. Examination of the captioned Application, as now amended, is respectfully requested.

Summary Of Telephone Interview: Appreciation is expressed for the opportunity on December 4, 2006, to review with the Examiner the reasons for the conclusions reached in response to the Request For Pre-Appeal Brief Request For Review. In that interview, the interpretation of the Apte reference by the Examiner was discussed, as follows. Apte was interpreted as teaching replication of Field 1 (1210) and Field 2 (1212) of FIG. 12 in the container 1204. The basis was that C17, L55+ were said to attribute to container 1204 the property of (L53+):

“...could persist fields 1210 and 1212 without assistance from
methods outside the container 1204”.

The Examiner interpreted the “without assistance...” text as a teaching that the container 1204 itself (on the server 1202) is the site of the replication of the storage of fields 1210 and 121, which is in addition to being the site of the original data to be replicated. As so interpreted, the replication of the field 1210 and 1212 is in the same container 1204 that is on the same server 1202. Moreover, this was interpreted as being a memory type of replication since the container 1204 is interpreted as being stored in memory via the server 1202.

As to the function of back-end storage 1222 of FIG. 12 of Apte, the storage 1222 was interpreted as teaching replication only to a disk type of state server. This replication is per C17, L58+ referring to tie object 1206 assisting in the persistence of EJB 1208, and C18, L1-6 was a basis for the replication of the referred EJB in back-end data store 1222.

In that interview, the interpretation by the Examiner of the claims was reviewed, as follows. Reference was made to Applicants’ FIG. 3. Although rejected claim 1 does recite a state object that stores state in the memory address space of the Java server process, and does define replicating that state, the “replicating” in rejected claim 1 was interpreted as follows. Such claim 1 was said to not clearly define that the state object (that stores a state within the memory address space of the Java server process) stores that state for replication in a state

server separate from the memory address space of the Java server process. Thus, such claim 1 was interpreted as not indicating that the state servers (that store a state to replicate that state) are separate from the memory address space of the Java server process, as is described in FIG. 3.

Although rejected claim 9 does recite partitioning and storing of a state in a state object, such claim 9 was interpreted as not clearly defining where such state object is stored, e.g., for the purpose of replicating that state. Thus, such claim 9 was interpreted as not indicating that the state servers (that store a state to replicate that state) are separate from the memory address space of the Java server process, as is described in FIG. 3.

Although rejected claim 18 does recite state objects that store state within the memory address space of the Java server process, and does recite replicating to state servers, such claim 18 was interpreted as not clearly defining where such state object is stored during replicating that state. Thus, such claim 18 was interpreted as not indicating that the state servers (that store state during replicating of that state) are separate from the memory address space of the Java server process, as is described in FIG. 3.

In view of these interpretations of the Apte reference by the Examiner, and in view of these interpretations by the Examiner of such claims, discussion in the interview was then directed to how to amend such claims so as to distinguish over the Examiner's interpretation of Apte. It is respectfully submitted that the present amendments to claims 1, 9, and 18 are consistent with the discussion with the Examiner.

Discussion of Amended Claims: It is respectfully submitted that the amended claims 1, 9, and 18 do now clearly define where such state objects are stored, and do now clearly define that the state servers (that store a state to replicate that state) are separate from the memory address space of the Java server process. In detail, the amended claims define the above distinguishing features as follows:

Claim 1: storing a plurality of state objects, each state object storing a state of a corresponding entity bean object within a memory address space of a Java server process,... replicating each one of the plurality of state objects in a state server, each of the state servers being separate from the memory address space of the Java server process.

Claim 9: partitioning individual entity bean objects... by storing state of each particular entity bean object in a state object dedicated to a state management type...., ... the recoverable state being one of a memory replicated state management type and a disk replicated state management type ..., ... each state object storing the state of a corresponding entity bean object within a memory address space of a Java server process; replicating each particular state management unit in one of a plurality of state servers....., each of the state servers being separate from the memory address space of the Java server process.

Claim 18: a plurality of state objects, each state object storing a state of a corresponding entity bean object within a memory address space of a Java server process, a state server dedicated to each state management type,each of the state servers being separate from the memory address space of the Java server process; ... a replicated state manager configured to replicate a particular state management unit to the state server...

In view of the interpretation by the Examiner of Apte as reviewed above, and the amendments to these claims 1, 9, and 18, the following is respectfully submitted. The basis of the above-noted interpretation of Apte (i.e., the container 1204 persists fields 1210 and 1212 without assistance from methods outside the container 1204) means that the asserted persistence is in the server 1202 of Apte at which the container 1204 is located. Thus, the Apte server 1202 that stores a replication of state to memory, is one and the same state server 1202 as is used for storing the initial image of the state object. The initial image is the one to be replicated. Thus, server 1202 does not meet the amended claim text requiring each of the state servers (i.e., 1222 of Apte) to be separate from the memory address space of the Java server process 1202. The reason is that server 1202 is not separate from the same server 1202.

Apte does not map to Applicants' FIG. 3, whereas these claims do map to FIG. 3, e.g., by the references to the memory address space of a Java server process (e.g., 204) and to the state servers (e.g., 212 and 214) that are separate from such memory address space of the Java server process.

Accordingly, it is respectfully submitted that the Apte reference, cited under 35 USC 102(e), does not teach all of the limitations of claim 1 and of the claims dependent on claim 1. This applies also to new claim 26. Claim 26 defines replication in the state server dedicated to the memory replicated state management type, such state server being separate from the memory address space of the Java server process. It is respectfully submitted that these claims are patentable over Apte.

Further, it is respectfully submitted that the Apte reference, cited under 35 USC 103(a) as the primary reference, with the secondary Chung reference, does not teach all of the limitations of claims 9 and 18, and of the respective claims dependent on claims 9 and 18, such that these claims are patentable over Apte in view of Chung. This applies also to new dependent claims 27 and 28.

Claim 27 defines the operation of replicating the particular state management units with more particularity. The replicating is of the state management unit that collects state objects corresponding to the memory replicated state management type of the corresponding entity bean objects. This replicating is in the state server for the memory replicated state management type, and that state server is separate from the memory address space of the Java server process.

Claim 28 defines the system with more particularity. The replicated state manager is further configured to replicate the particular state management unit that has classified the state objects associated with the memory replicated state management type. That replicated state manager replicates that particular state management unit to the state server that is dedicated to the memory replicated state management type. Such state server is defined as being separate from the memory address space of the Java server process.

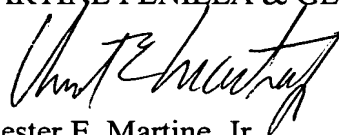
In review, an initial response was made to the Final Office Action by the Reasons For Review filed on 9/28/06. By the Pre-Appeal Brief Review process, the above-discussed issues of reference and claim interpretation were identified. Such Reasons were discussed in the interview, and resulted in the identification of the present amendments that distinguish over Apte in the manner identified above. In view of this, it is respectfully submitted that those issues are overcome by the amendments herein. Therefore, this Submission is submitted as the complete response required for purposes of 37 CFR 1.114 (c). Allowance of the pending claims is respectfully requested.

Application No. 09/825,249
Submission dated December 5, 2006
Response to Final Action Mailed July 17, 2006

In view of the foregoing, Applicants respectfully request allowance of the pending claims, as may be amended herein. Accordingly, a notice of allowance is respectfully requested.

The Examiner is requested to telephone Applicants' counsel at 408-774-6908 in the event that any questions arise in the examination.

Respectfully submitted,
MARTINE PENILLA & GENCARELLA, LLP



Chester E. Martine, Jr.
Reg. No. 19,711

710 Lakeway Drive, Suite 200
Sunnyvale, California 94085
(408) 774-6927
Customer Number 32291